

Implementación de un ERP de código abierto

Xavier Piernas Andrés

Resumen—El constante crecimiento del mundo empresarial demanda cada vez sistemas más avanzados, completos y eficientes para la gestión de los distintos ámbitos y procesos de negocio. Los clásicos programas de ordenador para la gestión básica de los pedidos y las facturas, acompañados de un cúmulo de hojas de cálculo, encuentran hoy en día sus limitaciones ante las necesidades del empresario moderno, cada vez más exigente e interesado en las nuevas tecnologías. Una versátil solución nace de esta necesidad y está viendo su crecimiento en la actualidad. Es el ERP, una potente herramienta de gestión que permite automatizar muchos procesos y agilizar las tareas del día a día de forma eficiente, ofreciendo una visión global del estado del negocio. Este proyecto muestra la implementación del ERP de código abierto Odoo y sus posibilidades de personalización para adaptarlo a las necesidades del negocio.

Palabras clave—ERP, Odoo, empresa, gestión

Abstract—The constant growth of the business world demands every time more advanced, complete and efficient systems for the management of the different areas and business processes. The classic computer programs for the basic orders and invoices management and the spreadsheets find their limitations today in front of the needs of the modern entrepreneur, who is every time more demanding and interested in new technologies. A versatile solution is born from this need and it is growing nowadays. It is the ERP, a powerful management tool which allows to automate many processes and make tasks faster efficiently, giving a global vision of the business state. This project shows the implementation of the open source ERP Odoo and its customization possibilities to adapt it to the business needs.

Index Terms—ERP, Odoo, business, management



1 INTRODUCCIÓN

Este proyecto tiene como finalidad la implementación del ERP Odoo al grupo de empresas Grup Espinaler.

Dado que la valoración de un proyecto de estas características es muy compleja, los responsables del grupo han solicitado a QubiQ la realización de una consultoría funcional con el objetivo, más allá de los requerimientos iniciales, de detallar los procesos y tareas a realizar para la implementación exitosa del proyecto. Este análisis más profundo debe servir para poder ofrecer una propuesta más concreta referente al alcance así como al coste de inversión a realizar.

Estos requerimientos se han ido recogiendo en las diferentes reuniones de trabajo realizadas con los distintos departamentos implicados en el proyecto, permitiendo así identificar cuáles de ellos se cumplen mediante parametrización de la herramienta o desarrollo a medida.

2 OBJETIVOS

Esta implementación tiene como meta sustituir la herramienta actual (Visual) y centralizar el resto de apli-

caciones de gestión del grupo en una única herramienta que ofrezca de forma unificada una visión del estado y ejecución de cada uno de los procesos de cada empresa del grupo.

Se desglosa esta meta en los siguientes objetivos:

- Unificar los distintos procesos de negocio en una única herramienta.
- Trasladar y mejorar la lógica de trabajo de los departamentos al nuevo sistema.
- Reducir mediante automatización y valores por defecto las tareas largas y repetitivas.
- Agilizar el circuito de negocio reduciendo el tiempo invertido en cada tarea.
- Conseguir una visión más global y precisa del estado del negocio.
- Lograr una implementación exitosa del ERP cubriendo todas las necesidades del negocio de forma más eficiente que el sistema anterior.
- Sacar el máximo partido posible a las funcionalidades estándar de Odoo, adaptando el cliente su forma de trabajo a la de la herramienta en la medida de lo posible, para lograr una forma de trabajo más eficiente y evitar conflictos.
- Ser realistas e implementar en la primera iteración sólo los requerimientos básicos e imprescindibles para el funcionamiento básico del negocio, dejando aquellos más complejos y de largo desarrollo para futuras iteraciones.

- E-mail de contacto: xavier.piernas@gmail.com
- Mención realizada: Ingeniería del Software
- Trabajo tutorizado por: Yolanda Benítez Fernández (Ciencias de la Computación)
- Curso 2017/18

3 ESTADO DEL ARTE

En los últimos años, numerosas empresas han decidido renovar su obsoleto sistema de gestión en busca de una herramienta que facilitara la realización de la multitud de tareas que un negocio implica y mejorara la eficiencia del mismo.

El ERP es una herramienta potencialmente apta para cualquier tipo de negocio, pero encuentra sus principales usuarios en las empresas de distribución y manufactura, según los datos recogidos en 2017 por "Panorama Consulting Solutions" en su informe anual sobre la elección e implementación de software de gestión entre las empresas de todo el mundo [1].

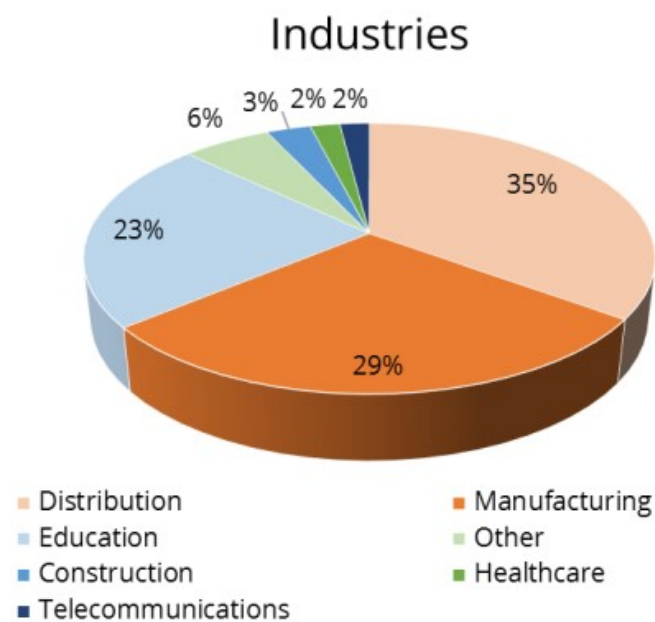


Ilustración 1. Implementación del ERP por sectores

La implementación de un ERP supone una gran inversión, pero también un aumento de la productividad logrado mediante la automatización de procesos, la integración de toda las aplicaciones en un mismo punto y el ahorro de tiempo y costes, siendo esta la principal razón por la que los empresarios optan por la implementación de la herramienta, como podemos observar en la encuesta realizada en el mismo informe [1].



Ilustración 2. Razones para implementar un ERP

Tradicionalmente, existen tres tipos de ERP, o más concretamente, implementaciones del mismo.

La implementación "on-premise" consiste en la instalación del software en los servidores locales y los ordenadores de la compañía y es administrado a nivel técnico por el personal de la misma.

Por otro lado, en la implementación "cloud" se aloja la herramienta en los servidores del proveedor y se accede a ella a través de un navegador web.

Finalmente, la implementación "SaaS" (Software as a Service) es similar a la anterior, con un alojamiento en la nube, pero se entiende el uso de la herramienta como la prestación de un servicio, por el cual se paga una suscripción.

La mayoría de empresas opta por la implementación "on-premise", dado que esta les ofrece un mayor control sobre el software y los datos, en contraste con la implementación "cloud", que es la menos demandada debido al riesgo de pérdida de datos y de brechas de seguridad [1].

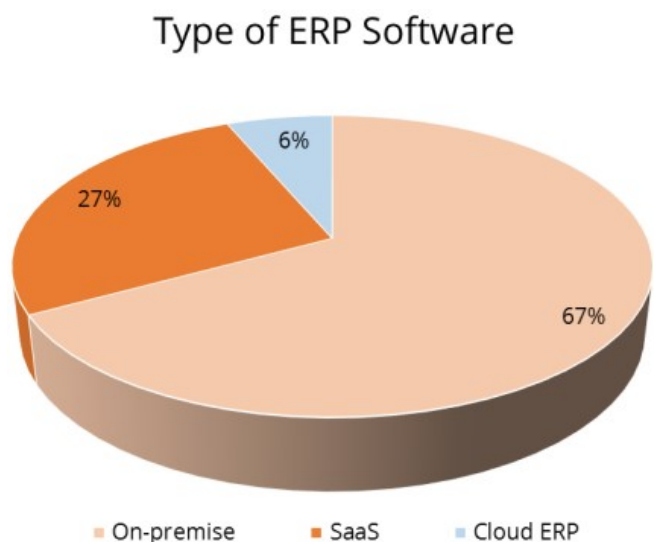


Ilustración 3. Tipos de implementación del ERP

Una de las principales ventajas de los ERP es que son altamente personalizables, pudiendo adaptarse de manera sencilla a las necesidades y demandas de cualquier negocio.

Aún así, es recomendable mantener un perfil de customización bajo y adaptarse lo máximo posible a la herramienta para evitar conflictos y trabajar de la forma más eficiente posible.

El nivel ideal de personalización a nivel de modificación de código está entre el 10% y el 20% y la mayoría de empresas se mueven dentro de ese rango [1].

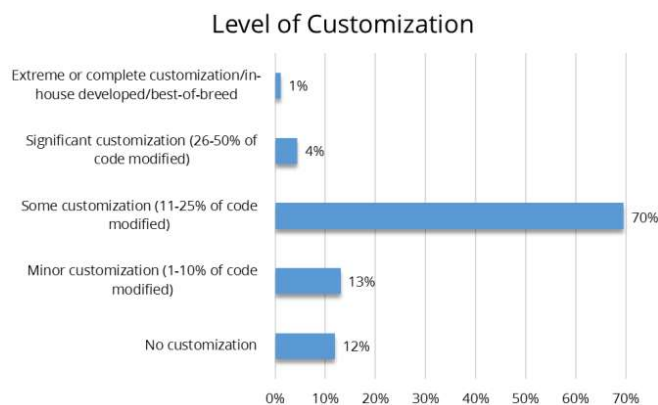


Ilustración 4. Nivel habitual de personalización

El incremento en la demanda de ERP tiene como consecuencia el incremento en la oferta. Cada vez más consultoras ofrecen sus servicios para la implementación, personalización y gestión de la herramienta.

El software de un ERP es denso y complejo y es altamente recomendable confiar a un “partner” especializado estas tareas.

Las razones más habituales por las que las empresas deciden contratar servicios externos para la implementación son la gestión profesional de la misma y el soporte en la gestión del cambio organizacional [1].

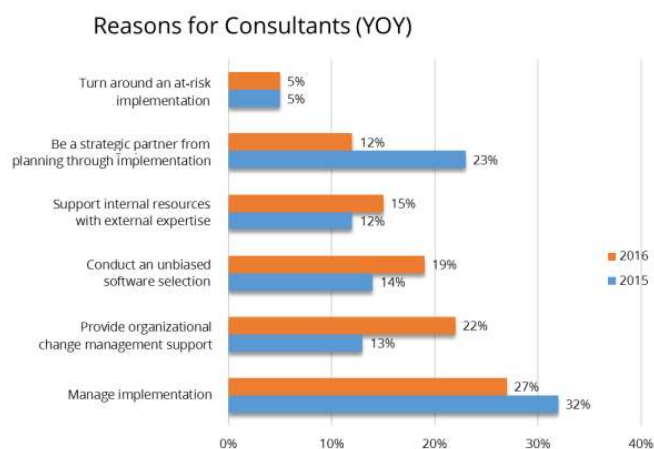


Ilustración 5. Razones para contratar un consultor

El proceso de migración y adaptación es algo complejo y costoso y suele ser el principal motivo de rechazo del cambio. Pero con esfuerzo, compromiso y dedicación por ambas partes, es altamente probable que el resultado sea exitoso.

Durante el último año, el 70% las implementaciones fueron llevadas a cabo con éxito [1].

Implementation Outcome

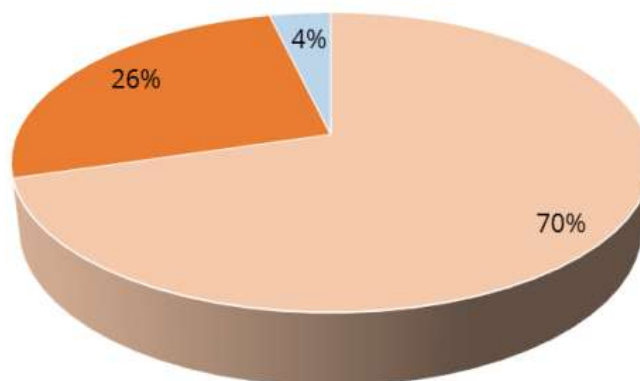


Ilustración 6. Resultados de implementaciones

Aún así, un 46% de las compañías reportó insatisfacción con respecto a su consultor y sólo un 25% afirmó estar satisfecho [1].

Overall Experience With ERP Vendors

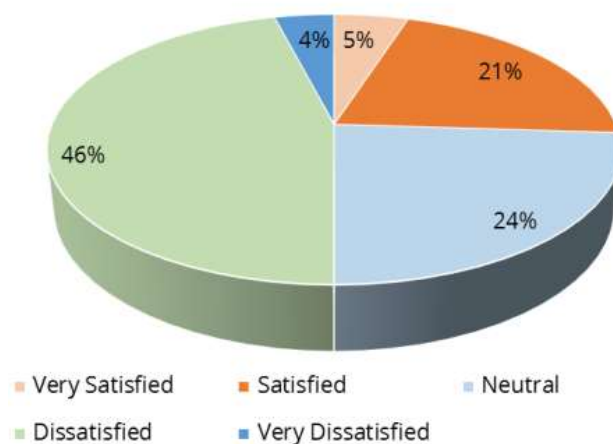


Ilustración 7. Satisfacción de las compañías

Es importante pues elegir un consultor de confianza y con experiencia para obtener unos resultados satisfactorios.

4 METODOLOGÍA

Durante la realización de este proyecto se ha hecho uso de la metodología de desarrollo en cascada. Este enfoque ordena rigurosamente las etapas del proceso de desarrollo de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.



Ilustración 8. Desarrollo en cascada

Se ha optado por esta metodología dadas las dependencias que existen en la mayoría de etapas del proyecto y con el objetivo de llevar a cabo una implementación secuencial, donde cada funcionalidad será analizada, implementada y probada de forma individual y completa y no se continuará hasta verificar su correcta integración en el sistema y descartar posibles conflictos con otras funcionalidades, siendo así más fácil identificar los errores y contextualizarlos de forma precisa y agilizando su resolución.

5 PLANIFICACIÓN

Dada la complejidad del proyecto y el tiempo que requiere, se ha decidido dividir su realización en dos fases principales. En este documento se trata la primera de ellas, que consiste en la completa implementación de los departamentos de ventas y compras, ya que son los que tienen una mayor implicación en el circuito empresarial y por tanto mayor relevancia en el proyecto.

5.1 Fases

Reuniones con el cliente. Reuniones varias con el director general y los diferentes departamentos.

Análisis. Análisis de los datos y requerimientos recogidos durante las reuniones.

Implementación del ERP. Implementación y configuración del ERP en el servidor.

Departamento de Ventas. Desarrollo e implementación del módulo de Ventas.

Departamento de Compras. Desarrollo e implementación del módulo de Compras.

Formación y resolución de incidencias. Documentación del proyecto, formación del cliente y resolución de incidencias.

Tabla 1. Fases del proyecto

5.3 Definición de fechas

Reuniones con el cliente

Definición de los objetivos	01/02 - 07/02
Captura de requerimientos	08/02 - 14/02
Planificación y definición de fechas	15/02 - 21/02
Análisis	01/03 - 16/03
Análisis de los requerimientos	01/03 - 06/03
Definición de las tareas a realizar	07/03 - 12/03
Asignación de prioridades	13/03 - 14/03
Asignación de tareas	15/03 - 16/03

Implementación del ERP

Configuración del servidor	19/03 - 20/03
Implementación del ERP	21/03 - 22/03
Instalación de los módulos básicos	23/03 - 26/03
Configuración funcional básica	27/03 - 28/03
Prueba del funcionamiento básico	29/03 - 30/03

Departamento de Ventas

Búsqueda de módulos	02/04 - 09/04
Instalación de módulos	10/04 - 11/04
Configuración de módulos	12/04 - 13/04
Prueba de módulos	16/04 - 17/04
Desarrollo personalizado	18/04 - 26/04
Prueba del desarrollo personalizado	27/04 - 30/04

Departamento de Compras

Búsqueda de módulos	01/05 - 09/05
Instalación de módulos	10/05 - 11/05
Configuración de módulos	14/05 - 15/05
Prueba de módulos	16/05 - 17/05
Desarrollo personalizado	18/05 - 29/05
Prueba del desarrollo personalizado	30/05 - 31/05

Formación y resolución de incidencias

Documentación del trabajo realizado	01/06 - 04/06
Formación del cliente en el uso de la herramienta	05/06 - 08/06
Uso y pruebas por parte del cliente	11/06 - 13/06
Resolución de incidencias	14/06 - 15/06

Tabla 2. Fechas clave del proyecto

5.3 Hitos

Reuniones con el cliente	Entrega de los requerimientos acordados	28/02/2018
Análisis	Entrega del análisis funcional	16/03/2018
Implementación del ERP	Entrega de la primera versión utilizable del ERP	30/03/2018
Departamento de Ventas	Entrega del módulo de Ventas	30/04/2018
Departamento de Compras	Entrega del módulo de Compras	31/05/2018
Formación y resolución de incidencias	Entrega de la documentación final	15/06/2018

Tabla 3. Entregables al cliente

6 ANÁLISIS

Este análisis de los requerimientos capturados trata de discernir y analizar aquellos requerimientos cuya implementación interesa llevar a cabo en esta primera fase, definiéndolos con mayor precisión y postergando el resto para la siguiente fase o futuras iteraciones.

6.1 Requerimientos funcionales

- ✓ **RF1.** El sistema usará como moneda únicamente el euro.
- ✓ **RF2.** El sistema ofrecerá los idiomas español y catalán para la interfaz.
- ✓ **RF3.** El sistema ofrecerá los documentos en español, catalán e inglés.
- ✓ **RF4.** El sistema elegirá el idioma de los documentos en función del idioma del cliente.
- ✓ **RF5.** El sistema permitirá modificar determinados datos de forma masiva.
- ✓ **RF6.** El sistema permitirá clasificar al cliente según su ámbito de negocio (mayorista, distribuidor, gran superficie, exportación...).
- ✓ **RF7.** El sistema permitirá realizar entregas parciales de los pedidos de venta.
- ✓ **RF8.** El sistema permitirá aplicar tarifas.
- ✓ **RF9.** El sistema no permitirá la creación de un cliente si no se rellena la dirección y el teléfono.
- ✓ **RF10.** El sistema permitirá o no modificar datos en función de los permisos de usuario.
- ✓ **RF11.** El sistema permitirá gestionar las comisiones de los vendedores.
- ✓ **RF12.** El sistema calculará el IBAN dada la cuenta corriente.
- ✓ **RF13.** El sistema permitirá gestionar los pedidos cancelados.
- ✓ **RF14.** El sistema permitirá gestionar los transportes.
- ✓ **RF15.** El sistema permitirá escribir comentarios en los pedidos de venta.
- ✓ **RF16.** El sistema permitirá gestionar los pedidos según las distintas líneas de negocio.
- ✓ **RF17.** El sistema no permitirá realizar pedidos que superen las existencias actuales.
- ✓ **RF18.** El sistema permitirá añadir múltiples productos al pedido de forma ágil.
- ✓ **RF19.** El sistema permitirá aplicar descuentos en los pedidos.
- ✓ **RF20.** El sistema ofrecerá información sobre las existencias en los pedidos de compra.
- ✓ **RF21.** El sistema permitirá gestionar los productos en función del proveedor.
- ✓ **RF22.** El sistema exigirá un responsable para validar las facturas de compra.
- ✓ **RF23.** El sistema permitirá aplicar descuentos sobre productos en las compras.

Tabla 4. Requerimientos funcionales

6.2 Requerimientos no funcionales

- ✓ **RNF1.** El sistema deberá permitir el acceso simultáneo a 100 usuarios.
- ✓ **RNF2.** El sistema deberá responder a cada solicitud en un tiempo inferior a 1 segundo.
- ✓ **RNF3.** El sistema deberá ser estable.
- ✓ **RNF4.** El sistema deberá ofrecer una disponibilidad superior al 98%.
- ✓ **RNF5.** El sistema deberá ser tolerante a fallos.
- ✓ **RNF6.** El sistema deberá ser seguro ante ataques.
- ✓ **RNF7.** El sistema deberá permitir realizar copias de seguridad cuando se desee.
- ✓ **RNF8.** El sistema permitirá aplicar tarifas.
- ✓ **RNF9.** El sistema deberá realizar copias de seguridad automáticas cada noche.

Tabla 5. Requerimientos no funcionales

6.3 Diagramas

Para comprender mejor el funcionamiento de los circuitos de negocio a tratar dentro del sistema de Odoo se han representado mediante unos sencillos casos de uso. Dejando a un lado acciones menos frecuentes y específicas como la creación y edición de productos y clientes se centran en el conjunto de acciones más habituales y que dan sentido a los ámbitos de interés, que son la creación de pedidos de venta y de pedidos de compra.

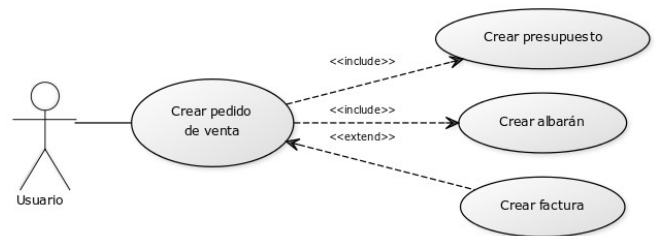


Ilustración 9. Caso de uso de venta

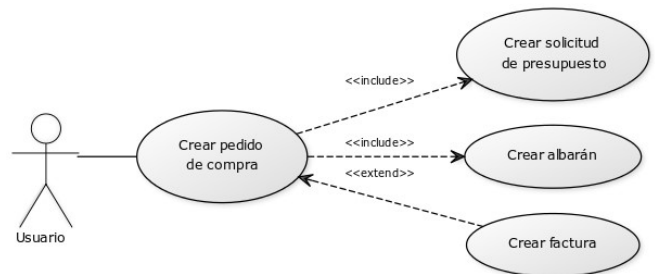


Ilustración 10. Caso de uso de compra

Como se puede observar, el concepto es idéntico en ambos casos. La creación de un nuevo pedido, ya sea por parte de un cliente hacia nosotros o nuestra hacia un proveedor, implica la existencia previa de un presupuesto. Una vez se confirma el presupuesto, este se convierte en el pedido como tal, lo que a su vez genera de forma automática un albarán asociado. Una vez creado el pedido, se tiene la opción de crear cuando nos interese una factura asociada.

A continuación, se profundiza en estos casos de uso mediante diagramas de secuencia que muestren el flujo de peticiones y respuestas entre el usuario y el sistema para ver con mayor detalle cada una de las acciones que el caso implica y sus consecuencias. Estos diagramas servirán en el futuro como guión para la realización de pruebas, verificando que los cambios realizados no afecten de una forma no deseada a los circuitos.

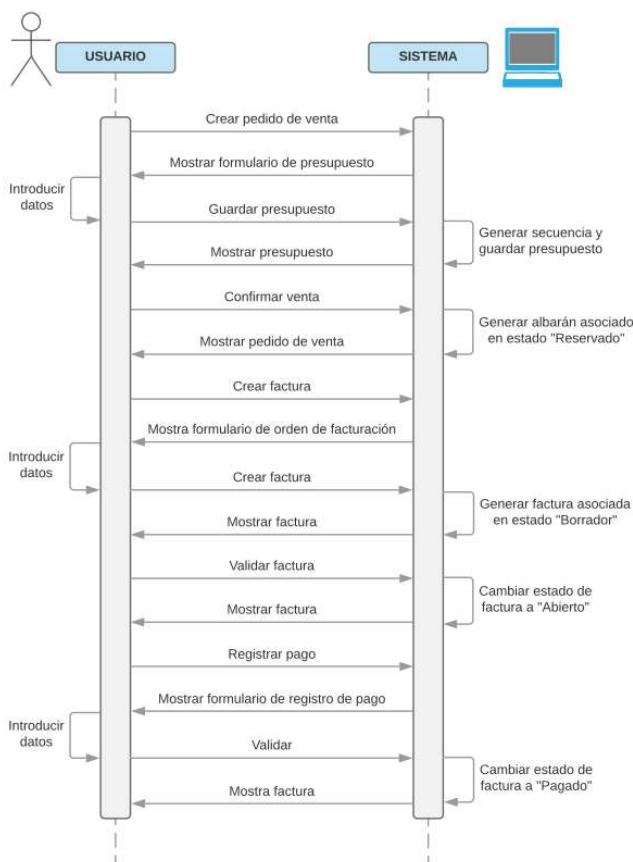


Ilustración 11. Diagrama de secuencia de venta

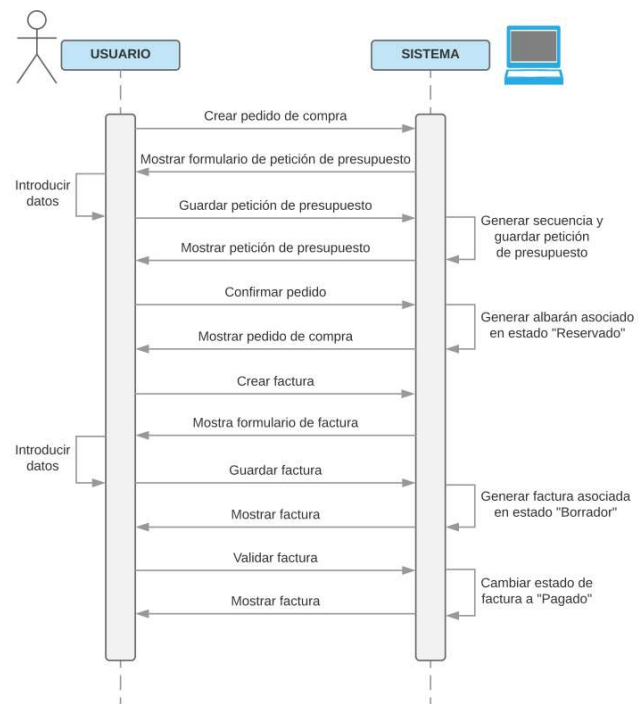


Ilustración 12. Diagrama de secuencia de compra

7 DESARROLLO

Se ha llevado a cabo la implementación y personalización de la herramienta Odoo y de los módulos de ventas y compras mediante parametrización de la herramienta, la instalación de múltiples módulos y alguna personalización particular de forma aislada, satisfaciendo un total de veintitrés requerimientos.

Se ha trabajado sobre la versión 10.0 de Odoo que se encuentra de forma gratuita en GitHub [2] y de forma local utilizando un contenedor Docker [3] sobre el sistema operativo Ubuntu y accediendo a la herramienta a través del navegador.

```
xavier@ASUS: ~/espinaler
xavier@ASUS:~$ cd espinaler
xavier@ASUS:~/espinaler$ docker-compose up
Creating espinaler_db_1 ... done
Creating odoo_espinaler ... done
Creating odoo_espinaler ... done
Attaching to espinaler_db_1, odoo_espinaler
db_1 | LOG: database system was interrupted; last known up at 2018-04-14 10:50:58 UTC
db_1 | LOG: database system was not properly shut down; automatic recovery in progress
db_1 | LOG: record with zero length at 0/67E0D58
db_1 | LOG: redo is not required
db_1 | LOG: MultiXact member wraparound protections are now enabled
db_1 | LOG: database system is ready to accept connections
db_1 | LOG: autovacuum launcher started
odoo_espinaler | 2018-04-14 10:51:28,758 1 INFO ? odoo: Odoo version 10.0-20171201
odoo_espinaler | 2018-04-14 10:51:28,759 1 INFO ? odoo: Using configuration file at /etc/odoo/odoo.conf
odoo_espinaler | 2018-04-14 10:51:28,759 1 INFO ? odoo: addons paths: ['/var/lib/odoo/addons/10.0', u'/mnt/custom-addons', u'/mnt/extra-addons', u'/usr/local/lib/python2.7/site-packages/odoo/addons']
odoo_espinaler | 2018-04-14 10:51:28,759 1 INFO ? odoo: database: odoo@db:
odoo_espinaler | 2018-04-14 10:51:28,770 1 INFO ? odoo.service.server: HTTP serv
```

Ilustración 13. Consola ejecutando el Docker

Tras crear la base de datos, instalar las aplicaciones básicas y establecer algunos datos de la compañía se obtiene un menú principal como este:

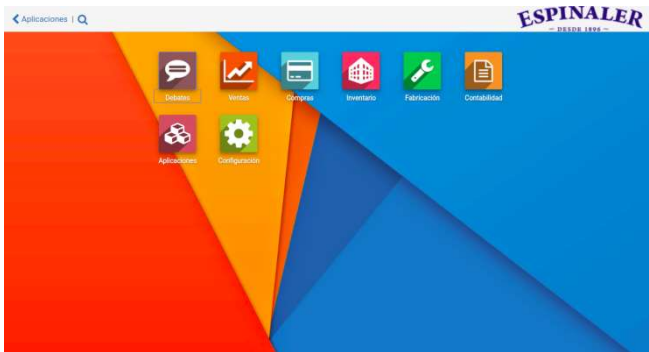


Ilustración 14. Menú principal de Odoo

Ocho de los requerimientos analizados se han satisfecho de manera estándar simplemente configurando la herramienta mediante las múltiples opciones de configuración que ofrece.

Precio

Precio de venta

- ☒ Un único precio de venta por producto
- ☐ Precios específicos por segmento de cliente, moneda, etc.
- ☐ Precios avanzados basados en fórmulas (descuentos, márgenes, redondeo)

Ilustración 15. Menú de configuración de Odoo

Por otro lado, se han instalado y probado un total de catorce módulos, la mayoría realizados por la comunidad OCA [4], listados a continuación:

- Mass Editing [5]
- Sales Commissions [6]
- Qu Iban Converter [7]
- Sale Cancel Reason [8]
- Stock Transport Management [9]
- Sale Comments [10]
- Sale Order Type [11]
- Deltatech No Negative Stock [12]
- Sale Product Multi Add [13]
- Sale Discount on Total Amount [14]
- Purchase Product Warehouse Quantity [15]
- Product by supplier info [16]
- Purchase order lines with discounts [17]

¡No hay suficiente inventario!

Usted planea vender 100.0 Unidad(es) pero solo tiene 50.0 Unidad(es) disponible!
El stock disponible es 50.0 Unidad(es).

ACEPTAR

Ilustración 16. Módulo "Deltatech No Negative Stock"

Por último, se han resuelto dos de los requerimientos mediante un desarrollo personalizado a nivel de código en el lenguaje de programación Python y trabajando con un patrón de modelos y vistas.

```
@api.multi
def action_invoice_open(self):
    super_bool = True
    if self.type == 'in_invoice' and self.company_id.needs_validation:
        if self.env.user.id == self.partner_id.validator_id or \
            self.env.user.has_group(
                'purchase_invoice_validation.purchase_invoice_validation_group'):
            if self.state == 'to_validate':
                self.state = 'draft'
            else:
                super_bool = False
                if self.state == 'to_validate':
                    raise UserError(
                        ("The invoice must be validated by a responsible!"))
                else:
                    self.state = 'to_validate'
    if super_bool:
```

Ilustración 17. Función del código desarrollado

8 RESULTADOS

Uno de los requerimientos desarrollados es el "RF22. El sistema exigirá un responsable para validar las facturas de compra".

Ahora al crear una nueva factura y tratar de validarla, pasa a un nuevo estado llamado "Pendiente de validación" y permanece ahí hasta que sea validada por el responsable elegido.

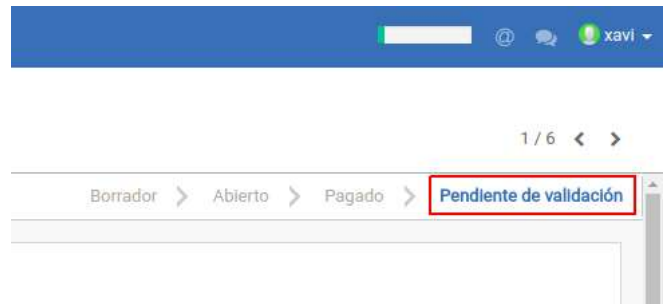


Ilustración 18. Factura pendiente de validación

Este responsable puede seleccionarse de forma individual para cada proveedor, como indicó nuestro cliente, mediante un nuevo desplegable en su formulario.



Ilustración 19. Validador de la factura

Como consecuencia del trabajo de análisis e implementación realizado, se ha obtenido una herramienta con un alto potencial de trabajo, adaptada a las necesidades de nuestro cliente, que mejora y agiliza el flujo de trabajo de la compañía.

9 PRUEBAS

La mayoría de módulos instalados contienen archivos de test para la comprobación del correcto funcionamiento de su funcionalidad en el entorno.

```
def setUp(self):
    super(TestMassEditing, self).setUp()
    model_obj = self.env['ir.model']
    self.mass_wiz_obj = self.env['mass.editing.wizard']
    self.mass_object_model = self.env['mass.object']
    self.res_partner_model = self.env['res.partner']
    self.ir_translation_model = self.env['ir.translation']
    self.lang_model = self.env['res.lang']

    self.partner = self.create_partner()
    self.partner_model = model_obj.\
        search([('model', '=', 'res.partner')])
    self.user_model = model_obj.search([('model', '=', 'res.users')])
    self.fields_model = self.env['ir.model.fields'].\
        search([('model_id', '=', self.partner_model.id),
              ('name', 'in', ['email', 'phone', 'category_id', 'comment',
                           'country_id', 'customer', 'child_ids',
                           'title'])])

    self.mass = self.create_mass_editing(self.partner_model,
                                         self.fields_model,
                                         'Partner')

    self.copy_mass = self.mass.copy()
    self.user = self.create_user()

    self.res_partner_title_model = self.env['res.partner.title']
    self.partner_title = self.create_partner_title()
    self.partner_title_model = model_obj.\
        search([('model', '=', 'res.partner.title')])
    self.fields_partner_title_model = self.env['ir.model.fields'].\
        search([('model_id', '=', self.partner_title_model.id),
              ('name', 'in', ['abbreviation'])])
    self.mass_partner_title = self.create_mass_editing(
        self.partner_title_model, self.fields_partner_title_model,
        'Partner Title')
```

Ilustración 20. Función de test

Para hacer uso de estos tests, se ha añadido un comando en el fichero que lanza el docker, haciendo que se ejecuten en el inicio.

```
ports:
  - "8069:8069"
environment:
  - DB10_PORT 5432 TCP_ADDR=db
  - DB10_ENV_POSTGRES_USER=odoo
  - DB10_ENV_POSTGRES_PASSWORD=odoo

command:
  - odoo
  - --config=/etc/odoo/odoo.conf
  - --test-enable

db:
  image: postgres:9.4
  environment:
    - POSTGRES_PASSWORD=odoo
    - POSTGRES_USER=odoo
    - PGDATA=/var/lib/postgresql/data/pgdata
```

Ilustración 21. Activación de los tests

Al iniciar el docker, podemos ver en la consola la ejecución de dichos tests y los resultados obtenidos.

```
xavier@ASUS: ~/espinaler
successful login from 'admin' using database 'espinaler'
odoo_espinaler | 2018-06-23 10:55:41,990 1 INFO espinaler werkzeug: 127.0.0.1 -
- [23/Jun/2018 10:55:41] "POST /xmlrpc/2/common HTTP/1.1" 200 -
odoo_espinaler | 2018-06-23 10:55:42,001 1 INFO espinaler odoo.addons.base.tests
.test_xmlrpc: test_xmlrpc_ir_model_search (odoo.addons.base.tests.test_xmlrpc.
test_xmlrpc)
odoo_espinaler | 2018-06-23 10:55:42,002 1 INFO espinaler odoo.addons.base.tests
.test_xmlrpc: Try a search on the object service.
odoo_espinaler | 2018-06-23 10:55:42,065 1 INFO espinaler werkzeug: 127.0.0.1 -
- [23/Jun/2018 10:55:42] "POST /xmlrpc/2/object HTTP/1.1" 200 -
odoo_espinaler | 2018-06-23 10:55:42,083 1 INFO espinaler werkzeug: 127.0.0.1 -
- [23/Jun/2018 10:55:42] "POST /xmlrpc/2/object HTTP/1.1" 200 -
odoo_espinaler | 2018-06-23 10:55:42,097 1 INFO espinaler odoo.addons.base.tests
.test_xmlrpc: Ran 2 tests in 0.208s
odoo_espinaler | 2018-06-23 10:55:42,097 1 INFO espinaler odoo.addons.base.tests
.test_xmlrpc: OK
odoo_espinaler | 2018-06-23 10:55:42,130 1 INFO espinaler odoo.modules.module: o
doo.addons.purchase.tests.test_purchase_order [running tests]
odoo_espinaler | 2018-06-23 10:55:42,130 1 INFO espinaler odoo.addons.purchase.t
ests.test_purchase_order: test_00_purchase_order_flow (odoo.addons.purchase.t
ests.test_purchase_order.TestPurchaseOrder)
odoo_espinaler | 2018-06-23 10:55:42,153 1 ERROR espinaler odoo.addons.purchase
.tests.test_purchase_order: ERROR
odoo_espinaler | 2018-06-23 10:55:42,160 1 INFO espinaler odoo.addons.purchase.t
```

Ilustración 22. Ejecución de los tests

Adicionalmente, se han realizado pruebas de forma manual tras la instalación de cada módulo, considerando cada aspecto afectado por el código y realizando todo el circuito implicado en la funcionalidad implementada.

10 CONCLUSIONES

Paralelamente a los objetivos particulares del proyecto, quería mostrar el enorme potencial de esta herramienta y su capacidad de adaptación a las necesidades particulares de cada negocio.

Me parece una herramienta a considerar y tener en cuenta por cualquier empresa, mediana o grande, que se encuentre actualmente utilizando un sistema antiguo y rudimentario que lastre el desarrollo de las tareas del día a día y entorpezca el flujo de trabajo.

Como se ha podido comprobar a lo largo del desarrollo, los únicos límites son la imaginación y la habilidad de los implantadores, y teniendo ambas se puede obtener todo lo que se desee, mejorando y agilizando en gran medida el trabajo a realizar.

Uno de los principales enemigos en este viaje de conversión es el miedo. Miedo a que la herramienta no cubra las necesidades de la forma esperada. Miedo a no saber usarla correctamente y que su uso sea un suplicio. Miedo a equivocarse y no poder reparar el error. Para vencer este miedo es importante mantener una relación de confianza y cordialidad con el equipo responsable de la implementación, sabiendo que va a estar dispuesto a ayudar ante cualquier duda o dificultad.

Una vez superado el proceso de formación y adaptación, trabajar será más cómodo que nunca. Porque si hay algo importante en un entorno de trabajo es sentirse cómodo y poder disfrutar de lo que haces, y sin duda con la nueva herramienta estaremos más cerca de lograr esta meta.

10.1 Seguimiento de la planificación

Se ha alcanzado con éxito y dentro de las fechas previstas el final de la implantación de los requerimientos acordados para la primera fase e iteración.

Se ha finalizado la personalización de los departamentos de ventas y compras mediante parametrización de la herramienta, la instalación de múltiples módulos y alguna personalización particular de forma aislada, satisfaciendo un total de 23 requerimientos.

No se han encontrado problemas, se han logrado los resultados esperados de forma satisfactoria.

10.2 Líneas abiertas

Dado que el proyecto se dividió en dos fases y hasta el momento sólo se ha tratado la primera, la línea de continuación inmediata será proceder a la segunda fase, que considerará ámbitos como contabilidad, fabricación e inventario.

Para ello se llevarán a cabo nuevas reuniones con los departamentos implicados para capturar nuevos requerimientos y se revisarán los requerimientos anteriormente capturados relativos a dichos ámbitos.

Finalmente, y una vez obtenida una versión de la herramienta completa y funcional, se estudiará la puesta en marcha de nuevas iteraciones para implementar requerimientos más complejos o de menor necesidad que se postergaron, así como nuevos requerimientos que surjan a raíz del uso de la herramienta.

AGRADECIMIENTOS

Quiero dar las gracias a mi tutora, Yolanda, por su atención y cordialidad a lo largo del desarrollo del proyecto, así como a mi familia y amigos por su apoyo y confianza.

BIBLIOGRAFÍA

- [1] "2017 Report on ERP systems & Enterprise Software", por Panorama Consulting Solutions.
<http://go.panorama-consulting.com/rs/603-UJX-107/images/2017-ERP-Report.pdf>
- [2] "Odoo", por Odoo S.A.
<https://github.com/odoo/odoo>
- [3] "Docker", por Docker, Inc
<https://www.docker.com/>
- [4] OCA (Odoo Community Association)
<https://odoo-community.org/>
- [5] "Mass Editing", por Serpent Consulting Services Pvt. Ltd., Tecnativa, brain-tec AG, Odoo Community Association (OCA)
https://www.odoo.com/apps/modules/10.0/mass_editing/
- [6] "Sales Commissions", por Odoo Community Association (OCA)
https://www.odoo.com/apps/modules/10.0/sale_commission/
- [7] "Qu Iban Converter", por QubiQ
<https://github.com/QubiQ/qu-server-tools>
- [8] "Sale Cancel Reason", por Camptocamp, Odoo Community Association (OCA)
https://www.odoo.com/apps/modules/10.0/sale_cancel_reason/
- [9] "Stock Transport Management", por Cybrosys Techno Solutions
https://www.odoo.com/apps/modules/10.0/stock_transport_management/
- [10] "Sale Comments", por Camptocamp, Odoo Community Association (OCA)
https://www.odoo.com/apps/modules/10.0/sale_comment_template/
- [11] "Sale Order Type", por Grupo Vermon, AvanzOSC, Tecnativa, Agile Business Group, Niboo, Odoo Community Association (OCA)
https://www.odoo.com/apps/modules/10.0/sale_order_type/
- [12] "Deltatech No Negative Stock", por Terrabit, Dorin Hongu
https://www.odoo.com/apps/modules/10.0/deltatech_stock_negative/
- [13] "Sale Product Multi Add", por ACSONE SA/NV, Odoo Community Association (OCA)
https://www.odoo.com/apps/modules/10.0/sale_product_multi_add/
- [14] "Sale Discount on Total Amount", por Cybrosys Techno Solutions
https://www.odoo.com/apps/modules/10.0/sale_discount_total/
- [15] "Purchase Product Warehouse Quantity", por Ascetic Business Solution
https://www.odoo.com/apps/modules/10.0/purchase_product_warehouse_quantity/
- [16] "Product by supplier info", por Elico Corp, Odoo Community Association (OCA)
https://www.odoo.com/apps/modules/10.0/product_by_supplier/
- [17] "Purchase order lines with discounts", por Odoo S.A., Acyos S.L., Tecnativa, ACSONE SA/NV, Odoo Community Association (OCA)
https://www.odoo.com/apps/modules/10.0/purchase_discount/

APÉNDICE

A1. DIAGRAMA WBS

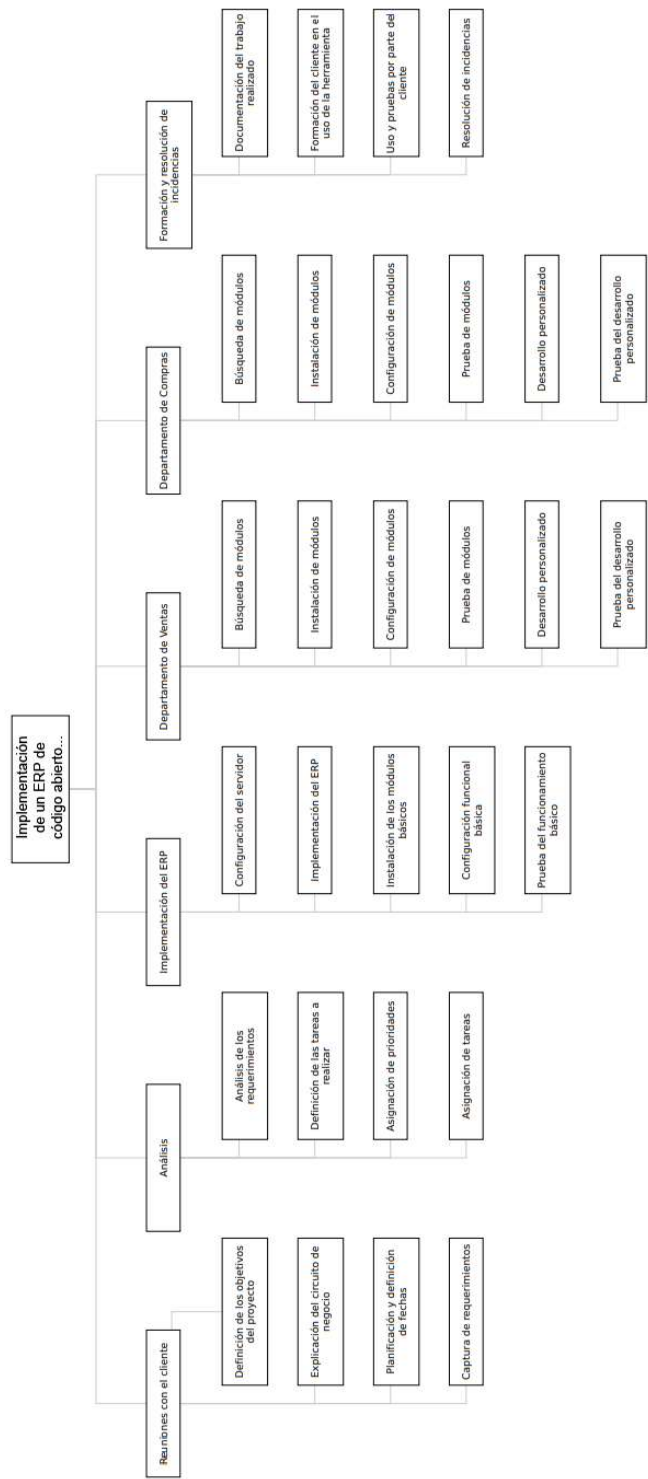


Ilustración 23. Diagrama WBS